UTD-CRSS SYSTEM FOR THE NIST 2015 LANGUAGE RECOGNITION I-VECTOR MACHINE LEARNING CHALLENGE

Chengzhu Yu, Chunlei Zhang, Shivesh Ranjan, Qian Zhang, Abhinav Misra, Finnian Kelly, John H. L. Hansen

Center for Robust Speech Systems (CRSS), Erik Jonsson School of Engineering, University of Texas at Dallas, Richardson, Texas, U.S.A.

{chengzhu.yu, john.hansen}@utdallas.edu

ABSTRACT

In this paper, we present the system developed by the Center for Robust Speech Systems (CRSS), University of Texas at Dallas, for the NIST 2015 language recognition i-vector machine learning challenge. Our system includes several subsystems, based on Linear Discriminant Analysis - Support Vector Machine (LDA-SVM) and deep neural network (DNN) approaches. An important feature of this challenge is the emphasis on out-of-set language detection. As a result, our system development focuses mainly on the evaluation and comparison of two different out-of-set language detection strategies: **direct out-of-set detection** and **indirect out-of-set detection**. These out-of-set detection strategies differ mainly on whether the unlabeled development data are used or not. The experimental results indicate that indirect out-of-set detection strategies used in our system could efficiently exploit the unlabeled development data, and therefore consistently outperform the direct out-of-set detection approach. Finally, by fusing four variants of indirect out-of-set detection based subsystems, our system achieves a relative performance gain of up to 45%, compared to the baseline cosine distance scoring (CDS) system provided by organizer.

Index Terms— language recognition, i-vector machine learning challenge, out-of-set detection, deep neural network

1. INTRODUCTION

The i-vector framework [1–10] has become the standard approach in state-of-the-art language recognition systems, due to its compact and efficient representation of language-dependent variability. Following the success of the i-vector machine learning challenge for speaker recognition [11–15], the motivation of the NIST 2015 i-vector machine learning challenge is to advance the classification systems used for language recognition [16]. By directly providing i-vectors to all participants, and thereby bypassing any front-end signal processing, the challenge is open to a broad range of researchers from different backgrounds.

The current i-vector challenge in language recognition poses a new challenge of detecting out-of-set languages without any labeled out-of-set samples. All participants are provided with labeled i-vector samples from 50 different languages, defined as in-set languages. Along with this labeled training dataset, an unlabeled development dataset is also provided. This development dataset includes both the samples from the 50 in-set languages as well as those belong to an unknown number of out-of-set languages.

This project was funded by AFRL under contract FA8750-12-1-0188 and partially by the University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by J.H.L. Hansen

By analyzing the official performance measure of this challenge, it can be observed that the weight associated with out-of-set language detection is significantly higher than any individual in-set language [16]. For example, missing one out-of-set language sample in the evaluation set results in a penalty that is more than 10 times higher than missing one in-set language sample. Therefore, accurate out-of-set language detection becomes the key to success in this challenge.

To address this new challenge of out-of-set language detection, we evaluate two different strategies: **direct out-of-set detection** and **indirect out-of-set detection**. The difference between these two approaches is on whether the unlabeled development data are used for out-of-set modeling or not. While both strategies are not new in the area of language recognition [17–19], the performance of those two approaches has been task dependent. Therefore it is important to evaluate both strategies in current i-vector machine learning challenge.

During our system development, both direct and indirect out-of-set detection approaches start by training an in-set language classification system on the labeled training data. LDA-SVM and DNN classifiers [20] are used for this purpose. In direct out-of-set detection approach, the out-of-set language samples in the evaluation data are detected by directly using threshold on the confidence scores obtained from in-set classification results. That is to say, we keep the in-set classification labels of evaluation data if the confidence score of classification is higher than a preset threshold, while treating the rest as out-of-set labels.

On the other hand, in indirect out-of-set detection approach, instead of directly applying threshold on the classification confidence scores in evaluation data, we first detect out-of-set language samples from the development dataset using the same confidence score based approach. After retrieving the out-of-set language samples from development data, the classifier is retrained by treating out-of-set as an additional language class. The retrained classifier with extended language classes are applied to evaluation data to achieve final results including both in-set and out-of-set labels. Based on the evaluation score on progress subset, we observe a significant advantage of indirect out-of-set detection approach compared to the direct out-of-set detection approach.

In addition to these out-of-set detection approaches, several important improvements on the baseline CDS based system are applied to achieve competitive overall results in the challenge. These improvements include: 1) additional out-of-set clustering using k-means; 2) score calibration for out-of-set detection in development data; 3) the use of duration as an auxiliary input feature along with LDA transformed features; 4) a majority voting based subsystem fusion.

2. LANGUAGE RECOGNITION SYSTEM

In this section, we describe two main classifiers used in our system both for in-set language classification and out-of-set language detection.

2.1. LDA-SVM subsystem

In this subsystem, the input i-vectors are first length normalized followed by a whitening transformation. Since the labeled training dataset was provided, it could be used to train an initial LDA transformation matrix with the purpose of maximizing the separability of 50 in-set languages, as well as dimensionality reduction. As the dimension of LDA must be less than the class number, the LDA matrix projects i-vector onto a 49-dimension vector. After obtaining out-of-set samples from development data during indirect out-of-set detection approach (see Sec. 3.2), the LDA is applied again to separate the out-of-set language from other languages. The LDA-transformed features are then fed into an SVM classifier both for in-set and out-of-set detection. An SVM with an RBF kernel is used in our experiments.

2.2. DNN subsystem

We trained a fully connected feed-forward neural network using the in-set training i-vectors for the DNN subsystem. 10% of the labeled training data per language (30 i-vectors) was randomly set aside as a *held-out* set to monitor the DNN training. The hidden-layer units used sigmoid activation function. The output layer used logistic regression nodes with a softmax function, with output nodes corresponding to the 50 in-set languages.

Next, we used the DNN to estimate the out-of-set labels from the development data using the scores of the output layer. We trained a second DNN with both the in-set and out-of-set labels that had an extra node in the output layer to also detect the out-of-set languages. The DNN had 2 hidden layers with 2048 nodes each. The second DNN was used in the language recognition experiments reported in this study. Further details on the design and implementation of the DNN based subsystem for language recognition can be found in [20].

3. OUT-OF-SET DETECTION

To address the challenge of out-of-set detection, we evaluated two different strategies for out-of-set detection: **direct out-of-set detection** and **indirect out-of-set detection**.

3.1. Direct out-of-set detection

The direct out-of-set detection approach is the most straightforward means of out-of-set detection. It can be achieved by simply applying confidence threshold on the classification scores obtained from the evaluation data. As initial classifier was trained with samples of 50 in-set languages, it can be assumed that an out-of-set language should have a low probability score output. In our experiment, we used LDA-SVM classifier for direct-out-set detection.

3.2. Indirect out-of-set detection

While the direct out-of-set detection achieves moderate performance improvement, the detection is based entirely on the knowledge of in-set languages. To achieve more discriminative training for out-of-set detection, a better way is to obtain some instances of out-of-set language, and then train a second classifier to discriminate out-of-set languages from other in-set languages. Specifically, our procedure for doing indirect out-of-set detection is as follows:

1. Perform i-vector length normalization and whitening transformation.
2. Train a 50-class in-set language classifier with training data.
3. Apply LDA-SVM as well as confidence thresholding approach to detect out-of-set language instances from development data.
4. Cluster the detected out-of-set language instances into multiple clusters using k-means algorithm.
5. Train a $50 + K$ classifier using both the training data and detected (and clustered) out-of-set language instances, where K is the number of k-means clusters from previous step.

The Fig. 1 is an illustrated distribution of detected out-of-set language samples from development data using indirect out-of-set detection strategy. In the following sections, we provide the details of the above steps.

3.2.1. confidence threshold

After training a 50-class LDA-SVM, the classifier is used to assign each sample of the development data into one of the 50 in-set class. We treat the probability of this LDA-SVM based in-set classification as the confidence of classification. Based on the fact that the classifier trained on in-set language samples would produce relatively lower probability on out-of-set languages, we use in-set classification confidence as a measure for detecting out-of-set instances from development data.

The main challenge with this confidence thresholding based approach is to find the appropriate threshold. In our experiment, we optimize this threshold from training data. Specifically, a randomly selected 30% of the training data is separated out and used as held-out dataset. The trained in-set language classifier is then applied to the held-out dataset. Based on the target and non-target classification probability distribution as in Fig. 2, the threshold for out-of-set detection is chosen as 0.4 in our system.

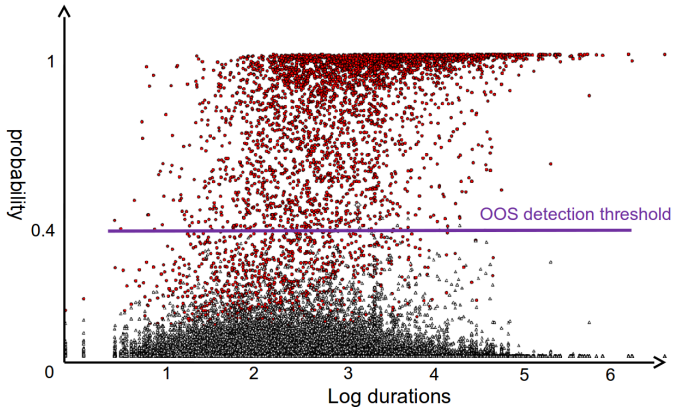

Fig. 2: *Red points show the distribution of target classification probablity as a function of log durations. Gray points show the distribution of non-target classification probability as a function of log durations. The purple line is the threshold for out-of-set detection.*

3.2.2. out-of-set clustering

To understand the problem of out-of-set detection more intuitively, we plot the distributions of each in-set language along with detected

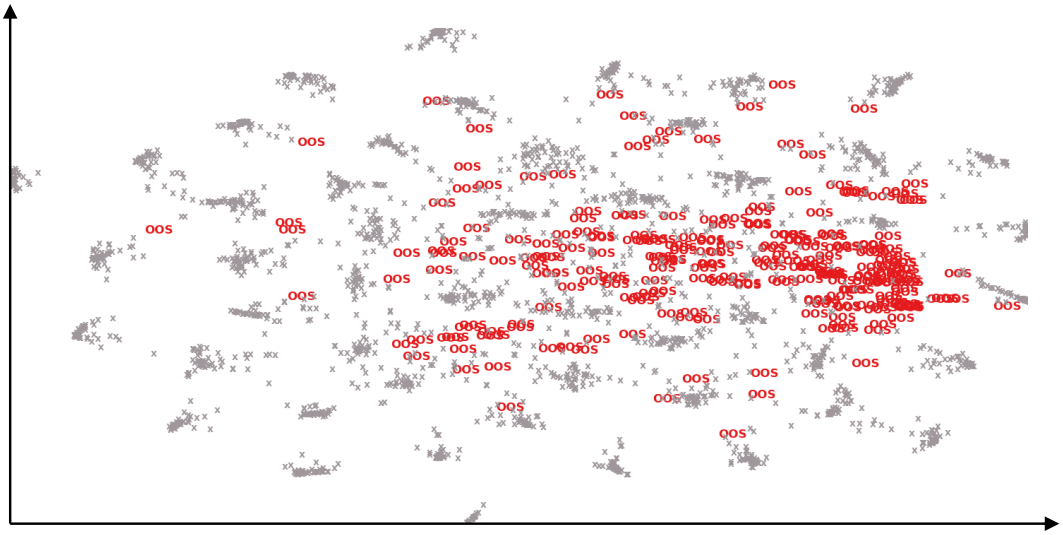

Fig. 1: *Ilustration of detected out-of-set language distribution with t-SNE scatter plot [21]. T-SNE is a superior method for visualizing high dimensional data in lower dimension. The gray marker 'x' indicates inset languages and the red marker 'oos' indicates out-of-set language instances detected from development data.*

out-of-set languages in Fig. 1. The first observation is that many detected out-of-set instances overlap with in-set clusters, indicating potential false detections. Another important observation is that the out-of-set languages are distributed over a larger area than any of the in-set languages, suggesting that they would be modeled best by using multiple classes. To achieve this, we further cluster the detected out-of-set languages into multiple clusters, using a bottom-up *k-means* algorithm.

3.2.3. score calibration

While the classification probability is a good indication of out-of-set language detection, the distribution of this probability varies for each language. Specifically, the distribution of target and non-target probability varies for each language and therefore the optimal confidence is class dependent. To reduce such score distribution variations between languages, we normalized the classification scores using the mean and variance of corresponding language score distribution.

3.2.4. class weighting

After extending the training data with detected out-of-set language instances, $(50 + K)$-class classifier is trained to classify both the 50 in-set languages as well as out-of-set languages at the same time. During the training of this $(50+K)$-class classifier, the 50 in-set languages are weighted equally, while higher weightings are applied to K out-of-set classes. This increased weighting on out-of-set classes is to achieve a low miss error rate on out-of-set detection.

3.2.5. durations

In our system, the duration information is used as an auxiliary feature for classifier training. The duration features is obtained simply by applying log function on the duration of each sample. The obtained duration feature is concatenated with LDA transformed features for training and classification.

4. EXPERIMENTS

In this NIST i-vector machine learning challenge, three datasets are released to participants by organizer. It includes training, development and evaluation datasets. The data in all three datasets comprises of i-vectors derived from previous NIST Language Recognitions Evaluations (LRE's) and other sources including IARPA BABEL Program. The duration of speech from which each i-vector was extracted is also provided. The training dataset is composed of 15000 instances of i-vectors belonging to 50 different languages, defined as in-set language in this challenge. All data in the training dataset comes with language labels. In addition, a separate development dataset of 6431 non-labeled i-vectors is also provided for training. The i-vectors in the development dataset include both the 50 in-set languages as well as an unknown number of out-of-set languages. For the evaluation of the system, 6500 i-vectors are provided as test dataset. A randomly selected 30% of the test dataset is used to monitor the progress on the online challenge leaderboard. The results we report in this paper are based on this dataset. The scoring metric used in this evaluation is the cost function defined by:

$$Cost = \frac{(1 - P_{OOS})}{n} * \sum_{k}^{n} P_{error}(k) + P_{OOS} * P_{error}(OOS). \quad (1)$$

In (1), $P_{error}(k) = (\frac{no.\ of\ errors\ for\ class\ k}{no.\ of\ trials\ for\ class\ k})$, $n = 50$, and $P_{OOS} = 0.23$. From the cost function provided above, it can be concluded that the out-of-set detection is very important for success in this challenge.

4.1. In-set Language Identification

To evaluate the different classifiers for language recognition challenge independent of out-of-set detection quality, the classifier is trained to output only 50 in-set languages. The result is shown in Table 1. It can be seen that the LDA-SVM produces best performance for in-set language identification scenario.

Table 3: Performance of variant indirect out-of-set detection based systems as well as the fused system.

classifier	OOS clustering	score calibration	duration feature	weighting	Score
LDA-SVM	y	n	y	y	**22.67**
LDA-SVM	y	y	n	n	23.38
LDA-SVM	y	y	y	n	24.05
DNN	y	n	n	n	26.56
Fused	-	-	-	-	***21.84***

Table 1: Evaluation of different classifiers for in-set language identification.

System	Score
CDS (baseline)	39.59
LDA+CDS	38.03
LDA+SVM	**35.32**
DNN	37.38

Table 2: Comparison of direct and indirect out-of-set detection.

System	Score
Direct (DNN)	32.71
Direct (LDA-SVM)	28.51
Indirect (DNN)	26.56
Indirect (LDA-SVM)	**22.67**

4.2. Results with Out-of-set Detection

Both the direct out-of-set detection and indirect out-of-set detection approaches are evaluated and compared in our experiments. The result in Table 2 compares the best single system based on direct out-of-set detection against the best single system based on indirect out-of-set detection.

In addition, three variants of indirect out-of-set detection systems were developed based on LDA-SVM classifier. Those three systems vary by the strategies used in indirect out-of-set detection as described in Table 3. The differences within those three LDA-SVM based systems and DNN based system provides complementary information during final system fusion. The performance of above individual systems is shown in Table 3.

The Table 4 shows how each sub-module in our indirect out-of-set clustering approach contributed to our best performing single system.

Table 4: Contribution of each submodule of indirect out-of-set detection system to the best performing single system based on LDA-SVM.

System	Score
LDA-SVM	25.64
+ Duration feature	24.72
+ OOS clustering	23.49
+ Class weighting	**22.67**

4.3. System Fusion

Finally, the output of the three LDA-SVM based subsystems and DNN based subsystem are fused together at the final language classification level. Given the labels from each system, a simple scheme based on majority voting (with one system taking precedence) was applied. The result is shown in Table 3.

5. CONCLUSION

In this study, we give a detailed description along with score analysis of the system developed by the Center for Robust Speech Systems (CRSS), University of Texas at Dallas, for the NIST 2015 language recognition i-vector machine learning challenge. The proposed system focuses primarily on the effective out-of-set detection strategy by comparing two approaches: direct out-of-set detection and indirect out-of-set detection. According to our experiments, the indirect out-of-set detection systems significantly outperform the direct out-of-set detection approach and is able to achieve good performance on the final evaluation of the challenge. The out-of-set language detection approaches evaluated for this challenge could potentially be beneficial to many real-world out-of-set detection problems.

6. REFERENCES

[1] N. Dehak, P. J Kenny, Réda Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[2] S. J. D Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–8.

[3] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 1695–1699.

[4] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction.," in *INTERSPEECH*, 2011, pp. 857–860.

[5] Pavel Matějka, Oldřich Plchot, Mehdi Soufifar, Ondřej Glembek, Luis Fernando D'haro Enríquez, Karel Veselỳ, František Grézl, Jeff Ma, Spyros Matsoukas, and Najim Dehak, "Patrol team language identification system for darpa rats p1 evaluation," 2012.

[6] R. Saeidi, K. Lee, T Kinnunen, T Hasan, B. Fauve, P. M. Bousquet, E. Khoury, P. L. S. Martinez, J. M. K. Kua, C. You, et al., "I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification," in *Proc. Interspeech*, 2013.

[7] C. Yu, G. Liu, S. Hahm, and J. H. L. Hansen, "Uncertainty propagation in front end factor analysis for noise robust speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 4017–4021.

[8] C. Yu, G. Liu, and J. H. L. Hansen, "Acoustic feature transformation using ubm-based lda for speaker recognition," *Proc. Interspeech, Singapore*, 2014.

[9] C. Zhang, G. Liu, C. Yu, and J. H. L. Hansen, "I-vector based

physical task stress detection with different fusion strategies," in *Interspeech*, 2015.

[10] C. Yu, A. Ogawa, M. Delcroix, T. Yoshioka, T. Nakatani, and J. H. L. Hansen, "Robust i-vector extraction for neural network adaptation in noisy environment," in *Interspeech*, 2015.

[11] C. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds, "The NIST 2014 speaker recognition i-vector machine learning challenge," .

[12] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, "Hierarchical speaker clustering methods for the NIST i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[13] S. Novoselov, T. Pekhovsky, and K. Simonchik, "STC speaker recognition system for the NIST i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 231–240.

[14] G. Liu, C. Yu, A. Misra, N. Shokouhi, and J. H. L. Hansen, "Investigating state-of-the-art speaker verification in the case of unlabeled development data," in *Proc. Odyssey speaker and language recognition workshop, Joensuu, Finland*, 2014.

[15] G. Liu, C. Yu, N. Shokouhi, A. Misra, H. Xing, and J. H. L. Hansen, "Utilization of unlabeled development data for speaker verification," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 2014, pp. 418–423.

[16] "The NIST 2015 language recognition i-vector machine learning challenge," `http://www.nist.gov/itl/iad/mig/upload/lre_ivectorchallenge_rel_v2.pdf`.

[17] P. Torres-Carrasquillo, E. Singer, T. Gleason, A. McCree, D. Reynolds, F. Richardson, and D. Sturim, "The MITLL NIST LRE 2009 language recognition system," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4994–4997.

[18] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic, and discriminative approaches to automatic language identification.," in *INTERSPEECH*, 2003.

[19] V. Prakash and J. H. L. Hansen, "In-set/out-of-set speaker recognition under sparse enrollment," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2044–2052, 2007.

[20] S. Ranjan, C. Yu, C. Zhang, F. Kelly, and J. H. L. Hansen, "Language recognition using deep neural network with very limited training data," *ICASSP*, 2016.

[21] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, pp. 85, 2008.